ROAD DAMAGE PREDICTION FOR INTELLIGENT TRANSPORTATION SYSTEM

A Mini Project Report

submitted by

VARSHA C(MES20MCA-2058)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

February 2022

DECLARATION

I undersigned hereby declare that the project report **ROAD DAMAGE PREDICTION FOR INTELLIGENT TRANSPORTATION SYSTEM**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of Mr.NOWSHAD C V, Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:

VARSHA C(MES20MCA-2058)

Date:28/02/2022

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **ROAD DAMAGE PREDICTION FOR INTELLIGENT TRANSPORTATION SYSTEM** is a bonafide record of the Mini Project work carried out by **VARSHA C(MES20MCA-2058)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

I pay my gratitude to the Almighty for his invisible help and blessing for the fulfillment of this work.At the outset i express my heartful thanks to my guide **Mr.Nowshad C V** Assistant professor,Dept. of Master of Computer Applications for his valuable guidance and supervision. I take this opportunity to express my profound gratitude to Ms.Priya JD, as my project coordinator for her valuable support, timely advise and strict schedules to complete our project. I also grateful to all my teaching and non-teaching staff for their encouragement, guidance and whole-hearted support. Last but not least, We gratefully indebted to my family and friends, who gave us a precious help in doing my project.

VARSHA C(MES20MCA-2058)

Abstract

Deep learning-based technology is a good key to unlock the object detection tasks in our real world. By using deep neural networks, we could break a problem that is dangerous and very time-consuming but has to be done every day like detecting the road state. This paper describes the solution using YOLO to detect the various types of road damage in the IEEE BigData Cup Challenge 2020. Our YOLOv5x based-solution is light-weight and fast, even it has good accuracy. We achieved an F1 score of 0.58 using our ensemble model with TTA, and it could be an adequate candidate for detecting real road damage in real-time.

Keywords:Deep Learning, Road Damage Dataset, YOLO

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Road damage detection is a crucial problem, and many kinds of researches have developed to break it in this challenge. As one of the deep-learning way, we used a YOLO-based solution to detect road damage. We eliminated the bounding boxes of useless classes in the dataset , every image is checked whether it has an object or not. We trained our models by splitting them into training dataset and validation dataset such as 5Fold cross validation, used various data augmentation options, for example, hue, saturation, value for HSV, image translation, image scale, mosaic, etc, therefore the input images are augmented to train the model.We used ensemble models that are trained with a dataset to predict the road damage.

1.1.1 Motivation

With the increase in world's population, there has been increasing load on the infrastructure.Roads have been flooded with the vehicular traffic. It has become increasingly difficult to manage this traffic. This is the prime motivation behind making a damage detection system. One of the increasing problems the roads are facing is worsened road conditions. Because of many reasons like rains, oil spills, road accidents or inevitable wear and tear make the road difficult to drive upon. Unexpected damages on road may cause more accidents. Also because of the bad road conditions, fuel consumption of the vehicle increases; causing wastage of precious fuel. Because of these reasons it is very important to get an alert of such bad road

conditions, Collect this information and provide an notification to users who travel through road.But there are various challenges involved in this. First of all there are various methods to get the information about the road conditions. Then this information must be collected and distributed to all the vehicles that might need this information. Lastly the information must be conveyed in the manner which can be understood and used by driver. We in this project try to design and build such a system. This system proposes a road damage prediction.It is to be designed and implemented for vehicles.

1.2 Objective

The main advantage of Intelligent Transportation System is to provide a smooth and safe movement of road transportation. It's also helpful in the perspective of environment friendliness to reduce carbon emission. It provides many opportunities for automotive or automobile industries to enhance the safety and security of their travelers. These methods provide a good effect for traffic damage prediction, and can establish a rapid and effective forecasting model. Most of these methods are the prediction and analysis of traffic flow parameters.

1.3 Report Organization

The project report is divided into four sections. Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

Deep learning-based technology is a good key to unlock the object detection tasks in our real world. By using deep neural networks, we could break a problem that is dangerous and very time-consuming but has to be done every day like detecting the road state.We could find out the road damage as monitoring the road surface. A human-based road damage monitoring system could be the first answer but not a perfect solution because it is affected by a different condition such as weather, speed of the vehicle, the complexity of the road, and the difference of criteria from the individual inspection.To detect road damage accurately, it is important to obtain three-dimensional (3D) depth images. However, to acquire such images, a dedicated vehicle must be used. As such, it is not possible to inexpensively and exhaustively inspect all roads. Under such circumstances, we think that it is worthwhile to consider methods that can comprehensively survey road surfaces at low costs, such as methods that rely on smartphones. For example, it is possible to acquire data by attaching a smartphone to a parcel-delivery service, postal service, or public vehicle. Although the accuracy of the model obtained in this research is not high when compared to the proposals that use highly accurate sensors, the proposed model is nevertheless effective for a preliminary and exhaustive inspection of all roads in a district before more expensivemethods are implemented. In other words, those roads that will require 3D depth images can be identified through preliminary inspections with a smartphone

Chapter 3

Methodology

3.1 Introduction

In our life, road structure is an essential component. We use public transport to commute almost every day and get a delivery service such as food, clothes or furniture through the road. From the perspective of autonomous driving technology, the road infrastructure has to be managed and kept to be maintaining as perfect as possible to get rid of uncertain obstacles on the road. But, how could we find out the road damage before it acts up? As monitoring the road surface, we could notice where is a problem on the road and prevent the accident. A human-based road damage monitoring system could be the first answer but not a perfect solution because it is affected by a different condition such as weather, speed of the vehicle, the complexity of the road, and the difference of criteria from the individual inspection. Thus, researchers have developed much more robust and accurate automatic road surface detectors through various methods. For example, using a probabilistic relaxation technique based on 3D information , combining 2D gray-scale image and 3D laser scanning data or implementing a deep learning-based model such as CrackNet . In this challenge, the dataset is gathered by a Smartphone based method and we evaluated with various scenarios using YOLO based on a deep learning-based algorithm. It is light-weight and fast in the object detection task, so it is available to improve the Smartphone-based model for detecting road damage. Deep learning-based technology is a good key to unlock the object detection tasks in our real world. By using deep neural networks, we could break a problem that is dangerous and very time-consuming but has to be done every day like detecting the road state. This paper describes the solution

using YOLO to detect the various types of road damage in the IEEE Big Data Cup Challenge 2020. Our YOLOv5x based-solution is light-weight and fast, even it has good accuracy. We achieved an F1 score of 0.58 using our ensemble model with TTA, and it could be an adequate candidate for detecting real road damage in real-time.

3.2 Modules

1.Admin

*Login

*View users details

*Add notification

*View feedback

*Track user

*Add and manage routs

2.User

*Register

*Login

*View profile

*Road quality alert

*feedback

*View route

3.3 Developing Environment

*OPERATING SYSTEM : WINDOWS 10

*FRONT END : HTML, CSS, JAVASCRIPT

*BACK END : Mysql

*SOFTWARES USED : Jetbrains Pycharm , Android Studio

*TECHNOLOGY USED : PYTHON,JAVA

*FRAME WORK USED : Flask

3.4 Work Flow

YOLO has a single neural network architecture, predicts a set of bounding boxes and class probabilities at a sitting for every test image. First of all, it divides the full image by several a grid with a specific size, and anchor boxes are generated in every grid of input image by predefined scale and size. Each anchor box predicts the objectness score, box center offset x, box center offset y, box width, box height, and class scores at one time in contrast to a two-stage detector. Thus, YOLO is an extremely fast end-to-end algorithm to detect the objects, and it is called a one-stage object detector. Also, the performance of YOLO has improved over the development of deep learning technology, so there are updated versions for improving the light-weight, inference speed, and accuracy.

We used YOLOv5 for this road damage detection challenge because it is state-of-the-art in the YOLO family for now. Also, YOLOv5 has useful components such as data augmentation, state-of-the-art activation functions, utilization of multi-GPU training, and a convenient manual. YOLOv5 uses CSPNet [17] as the backbone to extract the feature map from the image, and it has a Spatial Pyramid Pooling layer (SPP) for using various input image size and improving the robust. We eliminated the bounding boxes of useless classes in the dataset.We trained our models by splitting them into training dataset 80cross-validation, used various data augmentation options, for example, hue, saturation, value for HSV, image translation, image scale, mosaic, etc, therefore the input images are augmented, as shown in Fig. 1, to train the model. The default hyper-parameters are applied such as SGD optimizer, learning rate 0.01, momentum 0.937, and weight decay 0.0005. And,when we used 50 epochs and 32 batch sizes, the model was trained stably and showed steady performance.

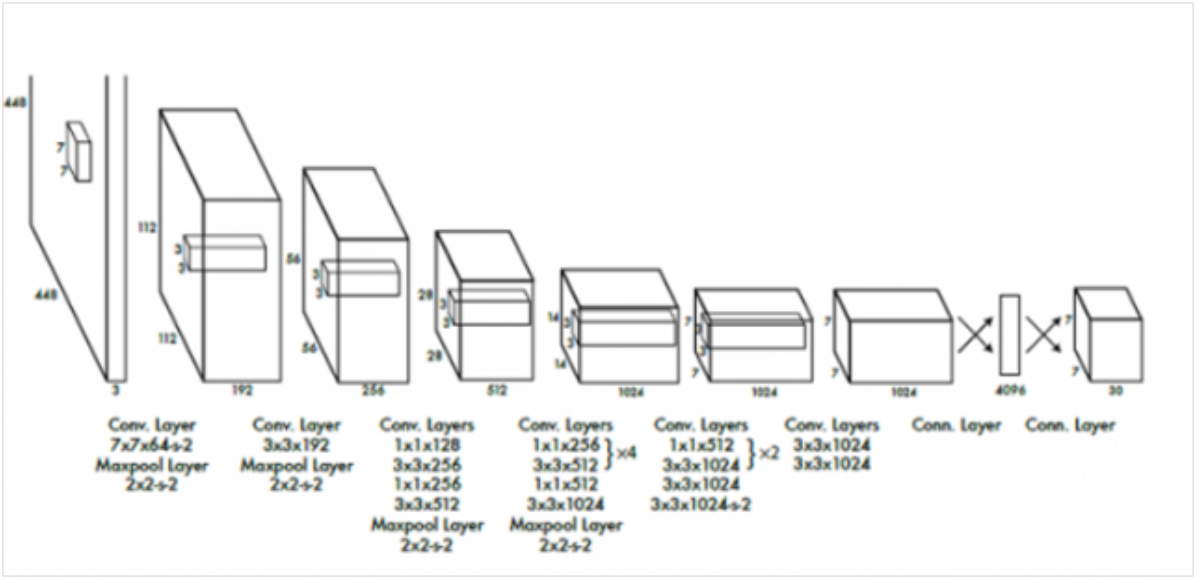

Figure 3.1: YOLO

3.5 User Story

A key component of agile software development is putting people first, and user-stories put actual end users at the center of the conversation. Stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building what they're building and what value it creates. A user story is a tool used in agile software development to capture a description of a software feature from an end- user perspective. The user story describes the type of user, what they want and why. A userstory helps to create a simplified description of a requirement. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work which drives collaboration, creativity, and a better product overall. The user story of system is given in Table 3.3

User Story ID	As a type of User	I want to <perform some task>	So that I can < Achieve Some Goal>
1	Admin	Login	login successful with correct username and password
2	Admin	View User details	Can view registered users
3	Admin	Add& manage notification	Insert ,view & remove notification
4	Admin	View feedback	Can view feedbacks from user
5	Admin	Track user	Track users who are travelling
6	Admin	Add and manage rout	Can add and manage rout
7	User	Register	Can users register
8	User	Login	Registered users can login with correct username and password
9	User	View profile	View users profile in application
10	User	Road quality alert	Can get alert while travelling
11	User	Feedback	Can provide feedback
12	User	View rout	Can view rout in application

Figure 3.2: User Story

3.6 Project Plan

A project plan that has a series of tasks laid out for the entire project, listing task durations,responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it. Project paln is given in Table 3.2

User Story ID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	26/12/2021	28/12/2021	10	completed
2		29/12/2021	31/12/2021		completed
3		03/01/2022	08/01/2022		completed
3	Sprint 2	09/01/2022	16/01/2022	13	Completed
4		18/01/2022	22/01/2022		Completed
5	Sprint 3	23/01/2022	27/01/2022	12	Completed
6		30/01/2022	05/02/2022		Completed
7	Sprint 4	06/02/2022	10/02/2022	9	Completed
8		16/02/2022	19/02/2022		Completed

Figure 3.3: Project Plan

3.7 Product Backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.The product backlog is the single authoritative source for things that a team works on. That means that nothing gets done that isn't on the product backlog. Conversely, the presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment.It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome. The Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories. The product backlog of the system is given in Table 3.3

User Story ID	Priority <High/Medium/Low>	Size (Hours)	Sprint <#>	Status <Planned/In progress/Completed>	Release Date	Release Goal
1	Medium	2	1	Completed	08/01/2022	Table design
2	High	3		Completed	08/01/2022	Form design
3	High	5		Completed	08/01/2022	Basic coding
3	High	5	2	Completed	22/01/2022	Data collection
4	Medium	5		Completed	22/02/2022	Data processing
5	High	5	3	Completed	05/02/2022	Prediction
6	medium	5		Completed	05/02/2022	Location based alert
7	Medium	5	4	Completed	20/02/2022	Testing data
8	High	5		Completed	20/02/2022	Output generation

Figure 3.4: Product Backlog

3.8 Sprint Backlog

Backlog Item	Status And Completion Date	Original Estimation in Hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory#1,#2,#3			hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Table Designing	28/12/2021	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Form Designing	31/12/2021	3	0	0	1	1	1	0	0	0	0	0	0	0	0	0
Coding	8/01/2022	5	0	0	0	0	0	1	1	1	1	1	0	0	0	0
UserStory#4, #5																
Data collection	16/01/2022	5	1	1	0	1	1	1	0	0	0	0	0	0	0	0
Data processing	22/01/2022	5	0	0	0	0	0	0	0	1	1	0	1	1	1	0
UserStory#6,#7																
Prediction	27/01/2022	5	1	1	1	0	1	1	0	0	0	0	0	0	0	0
Location based alert	5/02/2022	5	0	0	0	0	0	0	0	1	1	1	1	1	0	0
UserStory#8,#9																
Testing Data	10/02/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
OutputGenaration	20/02/2022	5	0	0	0	0	0	0	2	2	1	0	0	0	0	0
Total		40	4	4	3	3	4	3	3	5	4	2	2	2	1	0

Figure 3.5: Sprint Backlog

3.9 Actual sprint

Actual sprint backlog is what adequate sprint planning is actually done by project team there may or may not be difference in planned sprint backlog. The detailed sprint backlog (Actual) is given below.

Backlog Item	Status And Completion Date	Original Estimation in Hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory#1,#2,#3			hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Table Designing	28/12/2021	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Form Designing	31/12/2021	4	0	0	0	1	1	0	1	1	0	0	0	0	0	0
Coding	8/01/2022	4	0	0	0	0	0	0	0	0	2	1	1	0	0	0
UserStory#4,#5																
Data collection	16/01/2022	10	2	1	1	0	1	1	1	0	1	1	0	1	0	0
Data processing	25/01/2022	10	1	1	0	1	1	0	0	1	2	0	0	1	1	1
UserStory#6,#7																
prediction	27/01/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Location based alert	05/02/2022	5	0	0	0	0	0	0	0	1	0	1	1	1	0	1
UserStory#8,#9																
Testing Data	10/02/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Output Generation	19/02/2022	5	0	0	0	0	0	0	0	0	0	2	1	1	1	0
Total		50	6	5	3	4	5	1	2	3	5	5	3	4	2	2

Figure 3.6: Actual sprint

The project was developed using Agile development model. the entire project was divided into four sprints.In the first sprint Table design,form design and basic coding was developed.In the second sprint Data collection and data processing are completed.In third sprint prediction and location based alert was completed.In fourth sprint testing data and output generation was developed.

Chapter 4

Results and Discussions

4.1 Datasets

The image dataset is given by the IEEE BigData Cup Challenge 2020 for road damage detection, it consists of three countries such as Czech, India, and Japan. Each image is gathered by a smartphone application in JPEG format. The total number of images is 26,620 for training the road damage detector, 3,595 images from the Czech Republic, and 9,892 images from India were collected with a resolution of 720 x 720 pixels, and 13,133 images were captured with a resolution of 600 x 600 pixels in Japan. There are four damage types of pavement deterioration, such as D00 for the wheel- marked part, D10 for the equal interval, D20 for partial/overall pavement, and D40 for a pothole.

4.2 Results

We used ensemble models that are trained with a multi- country source dataset to predict the road damage for each country dataset with confidence threshold 0.4, NMS with IoU 0.5. Also, we applied Test-Time Augmentation (TTA) but it did not improve the performance of our model. We achieved F1 scores of 0.568 for the test1 dataset and 0.571 for the test2 dataset using our solution. After the challenge is finished, we performed several experiments to improve our model. For the Czech dataset, the 5 ensemble model using the Czech and 1,000 Japan images together showed the highest F1 score. If we need to consider inference speed too, the lower ensemble model could be a proper one for a real-time detector on the Czech road. And

from India test1 dataset, we could improve the F1 score up to 0.584 for the test1 dataset and 0.577 for test2 dataset using 5 ensemble models which are trained using only the India dataset with TTA.

Chapter 5

Conclusions

Road damage detection is a crucial problem, and many kinds of researches have developed to break it in this challenge. As one of the deep-learning way, we used a YOLO-based solution to detect road damage in the Czech Republic, India, and Japan. The dataset is collected by Smartphone applications from each country by 1 FPS. We evaluated various dataset scenarios using multi-country images within the Czech, India, and Japan, and it showed some interesting points. Using Japan road damage dataset with the Czech or India could affect the schedule of convergence of the model and generalization positively, but it does not always improve the performance of the model. For our YOLOv5x-based solution, one pre-trained weight needs just 170 MB memory and its inference speed is very fast. We achieved 0.584 for test1 dataset and 0.577 for test2 dataset with 5 or 6 ensemble models and TTA but could make a fast object detector also using lower ensemble and without TTA. In the perspective of real road damage detection problem, Not only accuracy but also inference speed is important even FPS can be a much more crucial point. Therefore, this solution could be an appropriate candidate for road damage detection on smartphone applications in real-time.

References

[1] **E. Salari and G. Bao** "Automated pavement distress inspection based on 2d and 3d information," in Electro/Information Technology (EIT). IEEE, 2011, pp. 1–4.

[2] **J. Huang, W. Liu, and X. Sun** , "A Pavement Crack Detection Method Combining 2D with 3D Information Based on Dempster-Shafer Theory," Comput. Aided Civ. Infrastructure Eng. (CACAIE), vol. 29, no. 4, pp. 299–313, 2014.

[3] **J.Redmon and A. Farhadi** , "YOLOv3: An Incremental Improvement," 2018. [Online]. Available: arXiv:1804.02767.

[4] **Alexey Bochkovskiy, Chien-Yao Wang and Hong-Yuan Mark Liao,** "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020. [Online]. Available: arXiv:2004.10934.

Appendix

Source Code

```
 from flask import *
from src.dbconnect import *
app=Flask(__name__)
app.secret_key="222"

import functools

def login_required(func):
   @functools.wraps(func)
   def secure_function():
      if "lid" not in session:
         return redirect("/")
      return func()

   return secure_function
@app.route('/')
def login():
   return render_template("login.html")


@app.route('/viewuser')
@login_required

def viewuser():

   qry = "SELECT * FROM user"
   s = select(qry)
   return render_template("viewuser.html",val=s)

@app.route('/notification')
@login_required

def notification():
   qry = "SELECT * FROM notification ORDER BY nid DESC"
   s = select(qry)

   return render_template("add notification.html",val=s)


@app.route('/addnotification',methods=['post','get'])
@login_required

def addnotification():
```

```
    notification=request.form['textarea']


    qry2 = "insert into notification values(null,curdate(),%s)"
    values = (notification)
    iud(qry2, values)
    return '''<script>alert('NOTIFICATION ADDED');window.location='/notification'</script>'''


@app.route('/addroutes',methods=['post','get'])
@login_required

def addroutes():
    froms=request.form['textfield']
    to=request.form['textfield2']
    route=request.form['textfield3']
    qry2 = "insert into route values(null,%s,%s,%s)"
    values = (froms,to,route)
    iud(qry2, values)
    return '''<script>alert('ROUTE ADDED');window.location='/manageroute'</script>'''


@app.route('/removoe')
@login_required

def remove():
    id=request.args.get('id')
    print(id)
    q="delete from notification where nid=%s"
    val=(id)
    iud(q,val)
    return '''<script>alert('deleted');window.location='/notification'</script>'''
@app.route('/delt')
@login_required

def delt():
    id=request.args.get('id')
    print(id)
    q="delete from route where id=%s"
    val=(id)
    iud(q,val)
    return '''<script>alert('deleted');window.location='/manageroute'</script>'''

@app.route('/addroute',methods=['post'])
@login_required

def addroute():
    return render_template("add route.html")


@app.route('/manageroute')
@login_required

def manageroute():
    qry="select * from route"
    res = select(qry)
    return render_template("manageroute.html",val=res)


@app.route('/viewfeedback')
@login_required

def viewfeedback():
```

Appendix

```
    qry="SELECT `user`.`fname`,`user`.`lname`,`feedback`.* FROM `feedback` JOIN `user` ON
          `user`.`user_lid`=`feedback`.`user_lid`"
    res=select(qry)
    return render_template("viewfeedback.html",val=res)

@app.route('/trackuser')
@login_required

def trackuser():
    qry="SELECT `user`.`fname`,`user`.`lname`,`user`.`phone`,`user`.`email`,`track`.* FROM `track` JOIN `user` ON
          `user`.`user_lid`=`track`.`user_lid`"
    res=select(qry)
    return render_template("trackuser.html",val=res)
@app.route('/homepage')
@login_required

def homepage():
    return render_template("homepage.html")




@app.route('/login2',methods=['post'])
def login2():
    uname=request.form['textfield']
    pword=request.form['textfield2']
    q="select * from login where username=%s and password=%s"
    val=(uname,pword)
    s=selectonecond(q,val)
    if s is None:
        return '''<script>alert('Invalid user name or password');window.location='/'</script>'''
    elif s[3]=='admin':
        session['lid']=s[0]
        return '''<script>alert('login successfully');window.location='/homepage'</script>'''




@app.route('/logout')
def logout():
    session.clear()
    return redirect('/')



app.run(debug=True)
```

Database Design

Attribute Name	Datatype	len	Description
id	Integer	11	Primary Key
username	Varchar	54	Unique
password	varchar	54	
usertype	varchar	54	

Table A.1: Login

Attribute Name	Datatype	len	Description
uid	Integer	11	Primary Key
user$_id$	integer	11	Unique
fname	varchar	55	
lname	varchar	33	
place	varchar	44	
phone	varchar	45	
email	varchar	45	

Table A.2: User

Attribute Name	Datatype	len	Description
id	Integer	11	Primary Key
imei	varchar	43	Unique
lattitude	varchar	43	
longitude	varchar	43	
image	varchar	43	
resultt	varchar	34	
datetime	varchar	43	

Table A.3: Alert

Attribute Name	Datatype	len	Description
id	Integer	11	Primary Key
userlid	integer	11	Unique
lattitude	varchar	54	
longitude	varchar	54	

Table A.4: Track

Attribute Name	Datatype	len	Description
id	Integer	11	Primary Key
from	varchar	55	Unique
to	varchar	45	
route	varchar	44	

Table A.5: Route

Attribute Name	Datatype	len	Description
fid	Integer	11	Primary Key
userlid	integer	11	Unique
date	date		
feedback	varchar	54	

Table A.6: Feedback

Attribute Name	Datatype	len	Description
nid	Integer	11	Primary Key
date	date		
notification	varchar	50	

Table A.7: Notification

DaTaflow Diagram

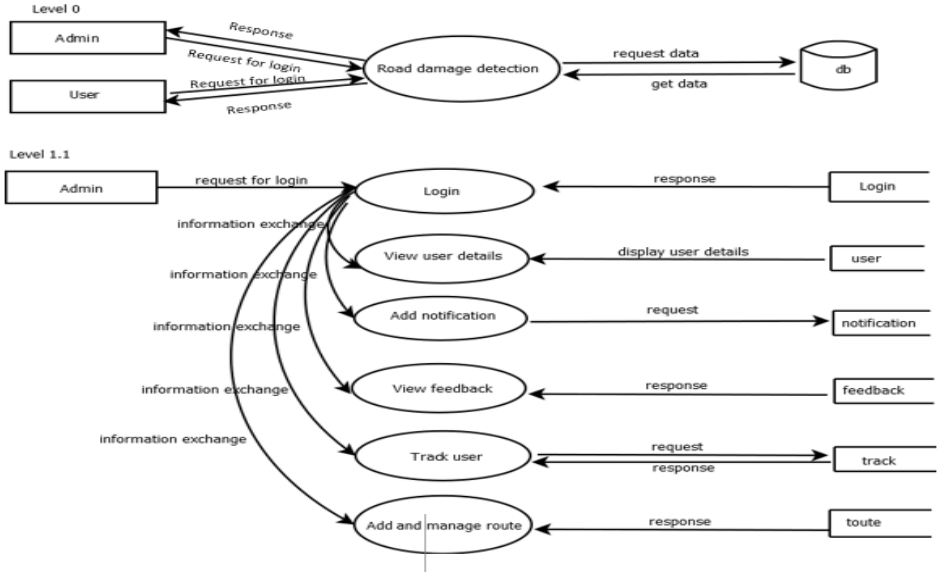

Figure A.1: Dataflow Diagram of admin

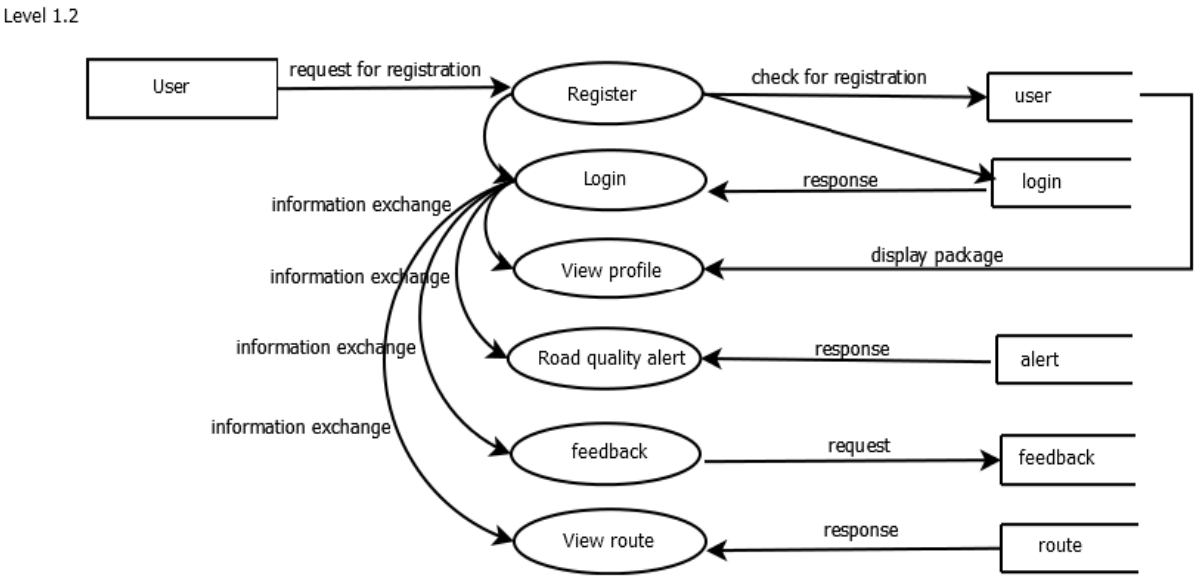

Figure A.2: Dataflow Diagram of user

5.1 User Interface

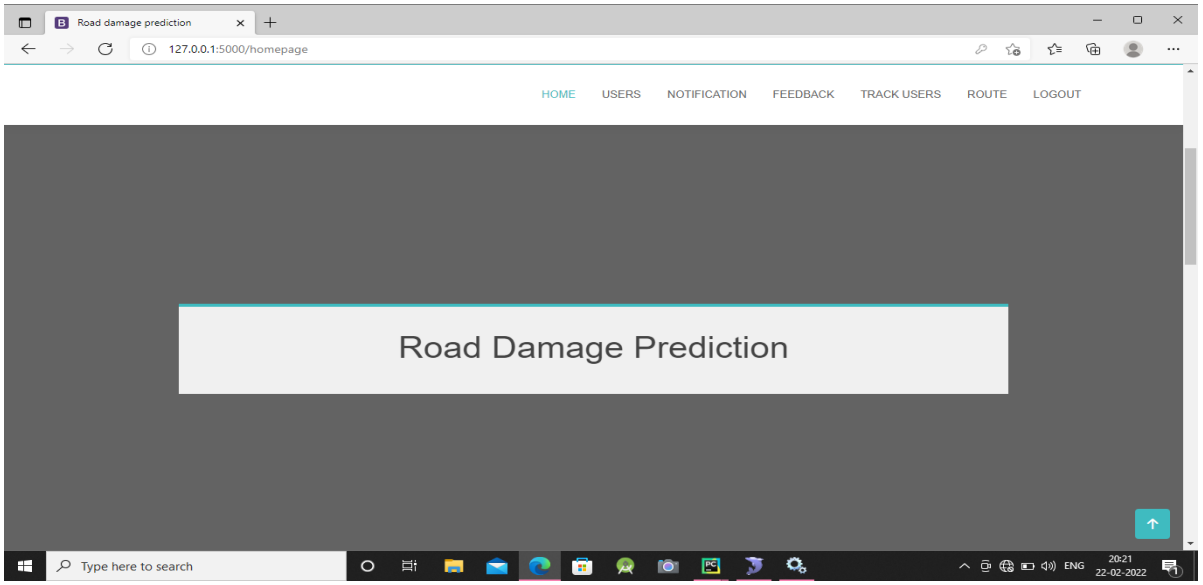

Figure A.3: Home page

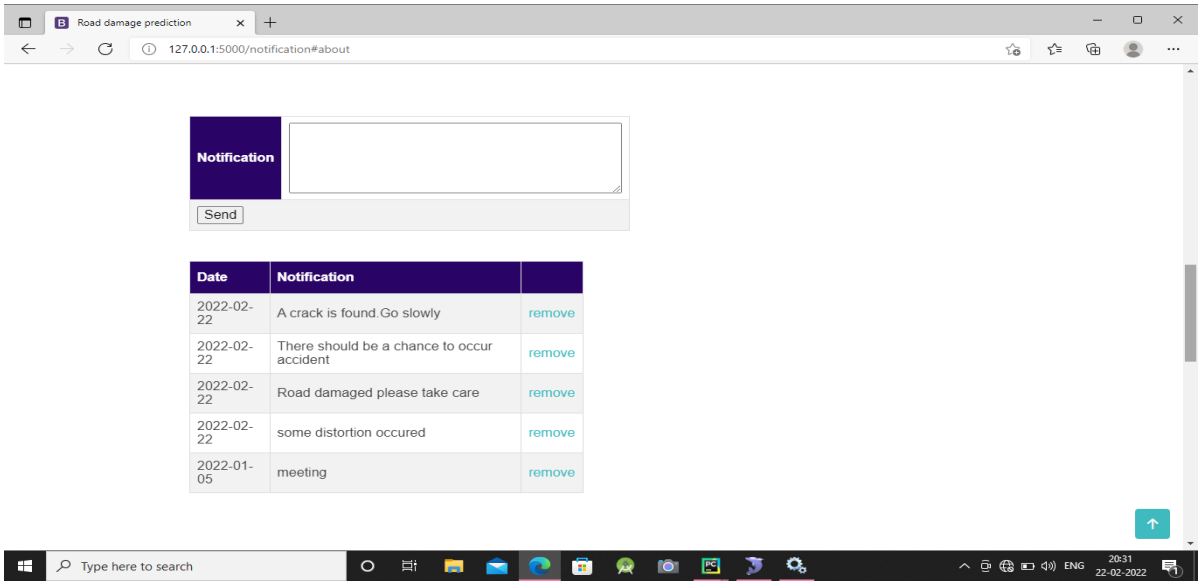

Figure A.4: Notification Page

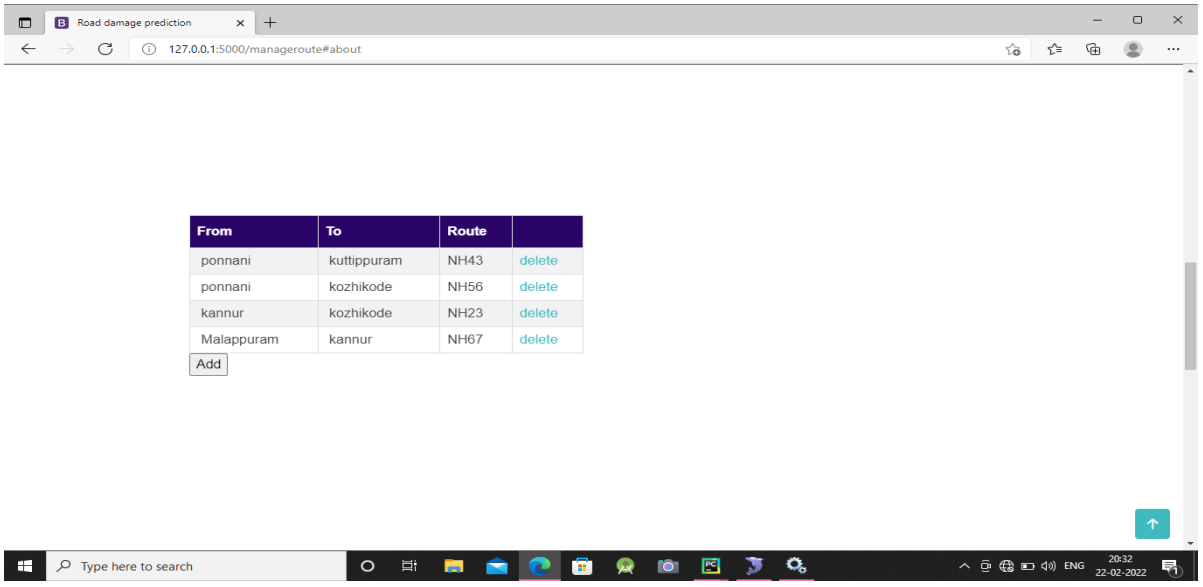

Figure A.5: Rout

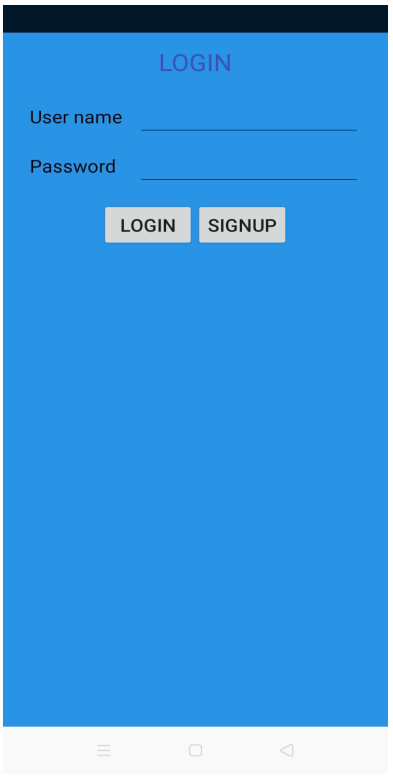

Figure A.6: Login

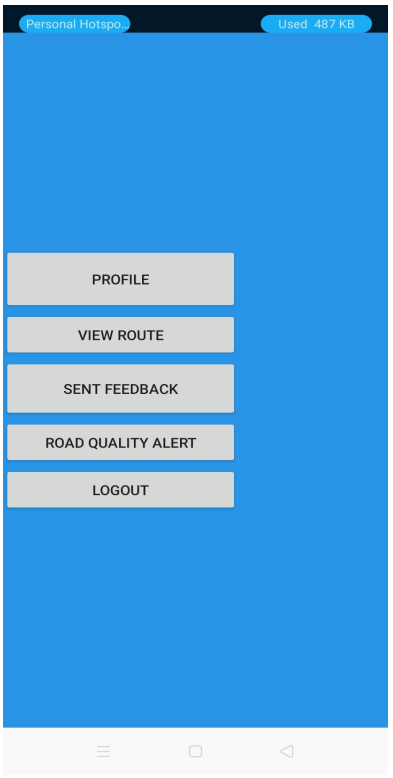

Figure A.7: Home page

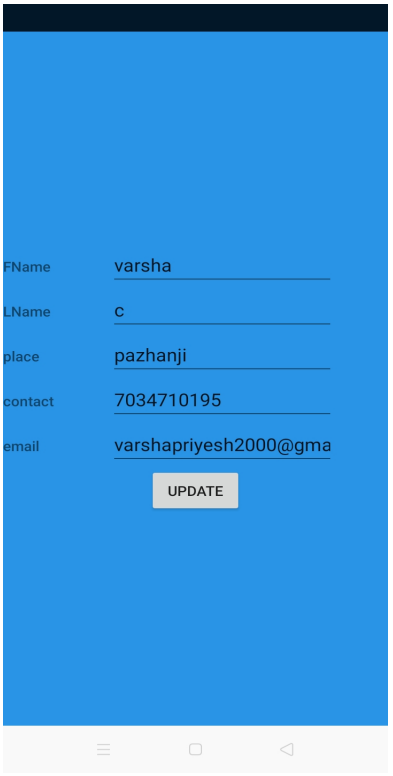

Figure A.8: Login